

#<sup>RS</sup>  
2

US

日本国特許庁

PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日  
Date of Application:

1999年 6月 9日

出願番号  
Application Number:

平成11年特許願第162702号

出願人  
Applicant(s):

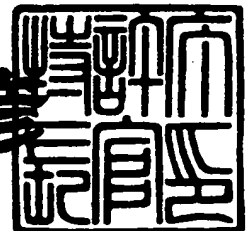
日本電気株式会社

JCS29 U.S. PTO  
09/588725  
06/07/00

2000年 3月10日

特許庁長官  
Commissioner,  
Patent Office

近藤隆彦



【書類名】 特許願

【整理番号】 66205984

【提出日】 平成11年 6月 9日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 9/46

【発明者】

    【住所又は居所】 東京都港区芝五丁目 7 番 1 号 日本電気株式会社内

    【氏名】 内堀 修作

【特許出願人】

    【識別番号】 000004237

    【氏名又は名称】 日本電気株式会社

【代理人】

    【識別番号】 100065385

    【弁理士】

    【氏名又は名称】 山下 穰平

    【電話番号】 03-3431-1831

【手数料の表示】

    【予納台帳番号】 010700

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 9001713

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 タスク処理システム

【特許請求の範囲】

【請求項 1】 発生したイベントの識別子を記憶する記憶装置と、前記イベントに基づいてタスクを生成するタスク制御装置と、前記タスクを処理するタスク処理装置とを備えたタスク処理システムであって、

前記タスク処理装置は、前記処理が完了したタスクと同一のタスクを生成する前記識別子を検索し、前記処理が完了したタスクと同一のタスクを更に処理することを特徴とするタスク処理システム。

【請求項 2】 前記検索の結果、前記処理が完了したタスクと同一のタスクを生成する前記識別子を見出さない場合には、前記処理が完了したタスクの資源の一部又は全部を前記記憶装置に開放することを特徴とする請求項 1 記載のタスク処理システム。

【請求項 3】 前記資源を、前記記憶装置から前記タスク制御管理装置を介して前記タスク処理装置に供給した場合には、前記資源を前記記憶装置から削除することを特徴とする請求項 2 記載のタスク処理システム。

【請求項 4】 前記記憶部は、前記タスク処理装置が実行処理中のタスクの識別子を記憶し、

前記タスク制御装置は、前記実行処理中のタスクと同一のタスクを生成する前記識別子を検索し、前記実行処理の完了を待って、前記実行処理の完了したタスクと同一のタスクを更に処理することを特徴とするタスク処理システム。

【発明の詳細な説明】

【0001】

【発明が属する技術分野】

本発明は、タスク処理システムに関し、特に、小規模なタスクを多量に処理する場合のタスク切り替えのためのオーバーヘッドの発生を抑制して、タスク処理速度を向上させるタスク処理システムに関する。

【0002】

【従来の技術】

従来、タスク処理システムにおいて、処理すべきタスクがあらかじめ分かっている場合には、あらかじめタスクをスケジューリングすることによりシステムのタスク処理速度を最適にする。

## 【0003】

一方、処理すべきタスクが予測不可能な場合には、割り込み等の非同期なイベントによりタスクが処理される。たとえば、特開平10-326197号公報に開示された「キュー処理方式」においては、ソフトウェアからの通常イベントと、タイマーやハードウェア割り込み等によるイベントを随時受付けて入力キーに登録し、割り込み禁止を行わずに、順次処理する。

## 【0004】

図5に示すように、上記公報に開示された「キュー処理方式」においては、具体的には、通常イベントキューには、割り込みイベントも含めてすべてのイベントのキューエントリーを用意する。割り込み1イベントが発生すると、その内容は、割り込み1イベント情報ブロックに格納され、そのイベントキューが、割り込み1イベントキューに追加される。割り込み2イベントが発生した場合も同様である。そして、割り込み1イベント情報ブロック及び割り込み2イベント情報ブロックに格納されたデータを、通常イベント情報ブロックに移動せずに、割り込みイベントキューの登録を解除し、通常イベントとして再登録する。

## 【0005】

## 【発明が解決しようとする課題】

しかし、従来の技術では、システム全体としてのタスク処理速度があがらない。その理由は、タスクをあらかじめスケジューリングすることが出来ないため、タスク処理に必要な資源の解放および確保のオーバーヘッドが発生するためである。

## 【0006】

又、従来の技術では、小規模なタスクを多量に処理することに向かない。その理由は、小規模なタスクを多量に処理する場合、タスク切り替えのオーバーヘッドが大量に発生するためである。

## 【0007】

そこで、本発明は、小規模なタスクを多量に処理する場合のタスク切り替えのためのオーバーヘッドの発生を抑制して、タスク処理速度を向上させることを課題としている。

#### 【0008】

##### 【課題を解決するための手段】

上記の課題を解決するための本発明は、発生したイベントの識別子を記憶する記憶装置と、前記イベントに基づいてタスクを生成するタスク制御装置と、前記タスクを処理するタスク処理装置とを備えたタスク処理システムであって、前記タスク処理装置は、前記処理が完了したタスクと同一のタスクを生成する前記識別子を検索し、前記処理が完了したタスクと同一のタスクを更に処理する。

#### 【0009】

すなわち、本発明においては、発生するイベントの種類に応じたタスクを処理するシステムにおいて、タスク処理完了後に同一種類のイベントが発生しているかどうかをチェックし、発生していた場合、同タスクを引き続き処理することにより、タスク処理に必要な資源の解放および確保のオーバーヘッドを減らし、システム全体のタスク処理速度を上げるようにしている。

#### 【0010】

##### 【発明の実施の形態】

以下、図面を参照して本発明の実施の形態について説明する。図1は、本発明のタスク処理システムのブロック図である。図1に示すように、本発明のタスク処理システムは、割り込み発生装置等のイベント発生装置1と、情報を記憶する記憶装置2と、プログラム制御等により動作するタスク処理装置4、および、発生したイベントに応じてタスク処理装置4を制御するタスク制御装置3とから構成されている。

#### 【0011】

記憶装置2は、イベント発生装置で発生したイベントを記憶するイベント記憶部21とプログラムおよびデータ等のタスク処理に必要な資源を記憶するタスク資源記憶部22とを備えている。イベント記憶部21に記憶されたイベントは、イベントの種類を識別するイベント識別子211を含む。

## 【0012】

タスク制御装置 3 は、イベントチェック手段 3 1 と、タスク生成手段 3 2 と、タスク資源制御手段 3 3 とを備える。

## 【0013】

イベントチェック手段 3 1 は、タスク処理装置 4 でタスク処理が完了すると、完了したタスクと同一のタスクを生成するイベントがイベント記憶部 2 1 に登録されているかどうかをチェックする。チェックには、イベント識別子 2 1 1 が使用される。

## 【0014】

タスク生成手段 3 2 は、イベント記憶部 2 1 に登録されたイベントに対応したタスクをタスク処理装置 4 で起動する制御を行う。イベントとタスクの対応付けには、イベント識別子 2 1 1 が使用される。

## 【0015】

タスク資源制御手段 3 3 は、タスク実行開始時の資源の確保、および、タスク実行完了時に資源の解放を行う。

## 【0016】

イベント記憶部 2 1 に登録されたイベントに応じたタスクがタスク生成手段 3 2 で生成され、タスク処理装置 4 で処理される。またこの時、タスク処理に必要な資源がタスク資源管理手段 3 3 によってタスク資源記憶部 2 2 から読み出され、タスク処理装置 4 に供給（以下、資源の確保と呼ぶ）される。

## 【0017】

タスク処理装置 4 でタスク処理が完了した時点で、イベントチェック手段 3 1 により、完了したタスクを生成したイベントと同一種類のイベントがイベント記憶部 2 1 に登録されているかが検査される。登録されていない場合、完了したタスクが必要とした資源のうち、タスク資源記憶部 2 2 への書き戻しの必要な資源がタスク資源管理手段 3 3 により戻される（以下、資源の解放と呼ぶ）。

## 【0018】

他の種類のイベントがイベント記憶部 2 1 に登録されている場合、タスク生成および資源の確保が繰り返される。完了したタスクを生成したイベントと同一種

類のイベントがイベント記憶部 21 に登録されている場合、資源の解放および再確保は行われず、引き続き、同一タスクがタスク処理装置 4 で処理される。

【0019】

このようにして、同一タスクを連続して実行することにより、異なるタスクを処理する際に必要な資源の解放および確保のオーバーヘッドを減らし、システム全体のタスク処理速度を向上させる。

【0020】

図 2 は、本発明のタスク処理システムの動作を説明するためのフローチャートである。イベント発生装置 1 で発生したイベントは、イベント記憶部 21 に記憶される。記憶には、イベントの種類を示すイベント識別子 211 が含まれる。

【0021】

まず、イベントチェック手段 31 は、イベント記憶部 21 に特定の識別子を持つイベントが記憶されているかどうかをチェックする（ステップ A1 および A2）。

【0022】

イベントが記憶されていれば次に、タスク資源管理手段 33 は、イベント識別子 211 に対応するタスクに必要な資源をタスク資源記憶部 22 から読み出し、タスク処理装置 4 に供給する（ステップ A3）。

【0023】

さらに、タスク生成手段 32 は、タスク処理装置 4 でイベント識別子 211 に対応するタスクを起動する（タスクを生成する）。この時、イベント記憶部 21 からは対応するイベントの記憶が削除される（ステップ A4）。

【0024】

イベントチェック手段 31 は、イベント記憶部 21 に完了したタスクに対応するイベント識別子 211 と同じ識別子を持つイベントが記憶されているかをチェックする（ステップ A5 および A6）。

【0025】

同じ識別子を持つイベントが記憶されている場合には、ステップ A4 に戻って、完了したタスクと同じタスクがタスク生成手段 32 により生成され、イベント

記憶部 21 からは対応するイベントの記憶が削除される。

【0026】

最後に、同じ識別子を持つイベントが記憶されていない場合、タスク資源管理手段 33 は完了したタスクの使用した資源で解放が必要な資源を、タスク資源記憶部 22 に書き戻す（ステップ A7）。

【0027】

図 3 は、本発明の他のタスク処理システムのブロック図である。図 3 の実施形態が図 1 の実施形態と異なる点は、タスク処理装置 4 を、タスク処理装置 41、42、4N のように複数台とした点と、記憶装置 2 に、実行タスク記憶部 23 を追加した点である。

【0028】

実行タスク記憶部 23 は、タスク処理装置 41、タスク処理装置 42、およびタスク処理装置 4N で実行中のタスクを記憶する。実行中タスク記憶部 23 に記憶されたタスクは、タスクの種類を識別するイベント識別子 231 を含む。

【0029】

図 4 は、図 3 の本発明の他のタスク処理システムの動作を説明するためのフローチャートである。図 4 には、タスク処理装置の個数に応じた N 個のフローチャートが示されているが、これらは、同一のものであり、それぞれ並列に動作する。

【0030】

図 4 のステップ B31 から B61 で示される本実施例におけるイベントチェック手段 31、タスク生成手段 32 およびタスク資源管理手段 33 の動作は、図 1 に示された実施例の各手段 31、32 および 33 の動作と同一である。

【0031】

イベント発生装置 1 で発生したイベントは、イベント記憶部 21 に記憶される。記憶には、イベントの種類を示すイベント識別子 211 が含まれる。

【0032】

まず、イベントチェック手段 31 は、イベント記憶部 21 に実行中タスク記憶部 23 に記憶されているタスクのイベント識別子 231 以外の識別子を持つイベ



ントが記憶されているかどうかをチェックする（ステップB 1 1 およびB 2 1）

。

#### 【0033】

ステップB 4 1で、タスク生成手段3 2は、タスク処理装置4でイベント識別子2 1 1に対応するタスクを起動する（タスクを生成する）。この時、イベント記憶部2 1からは対応するイベントの記憶が削除されると同時に、実行中タスク記憶部2 3に記憶される。

#### 【0034】

最後に、ステップB 7 1で、同じ識別子を持つイベントが記憶されていない場合、タスク資源管理手段3 3は完了したタスクの使用した資源で解放が必要な資源を、タスク資源記憶部2 2に書き戻す。また、実行中タスク記憶部2 3から対応するタスクの記憶が削除される。

#### 【0035】

##### 【発明の効果】

以上説明した本発明によれば、システム全体のタスク処理速度を向上できる。その理由は、タスク処理に必要な資源の解放および確保のオーバーヘッドを削減しているからである。

#### 【0036】

又、本発明によれば、規模の小さなタスクを多量に処理できる。その理由は、規模の小さなタスクを多量に処理する場合には特に、タスク処理の切り替えが発生しても、タスク処理に必要な資源の解放および確保のオーバーヘッドが削減されているため高速にタスクを処理することができるからである。

##### 【図面の簡単な説明】

##### 【図1】

本発明の第1の実施形態のタスク処理システムのブロック図

##### 【図2】

第1の実施形態のタスク処理システムの動作を説明するためのフローチャート

##### 【図3】

本発明の第2の実施形態のタスク処理システムのブロック図

【図 4】

第 2 の実施形態のタスク処理システムの動作を説明するためのフローチャート

【図 5】

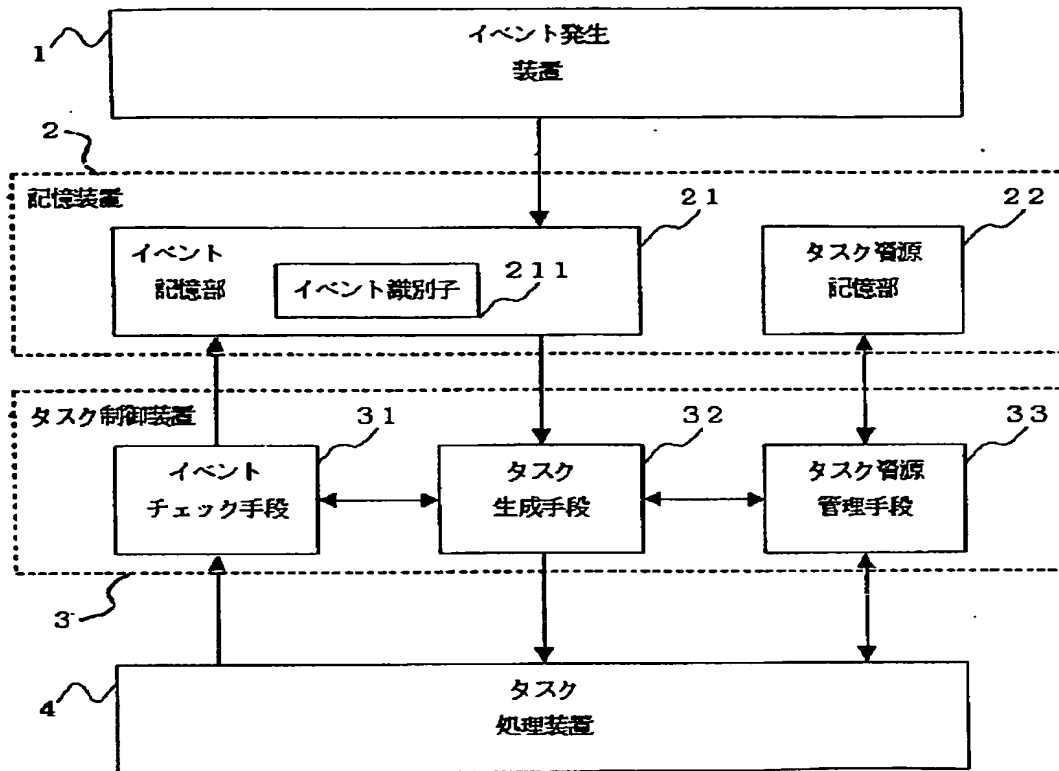
従来のキュー処理方式

【符号の説明】

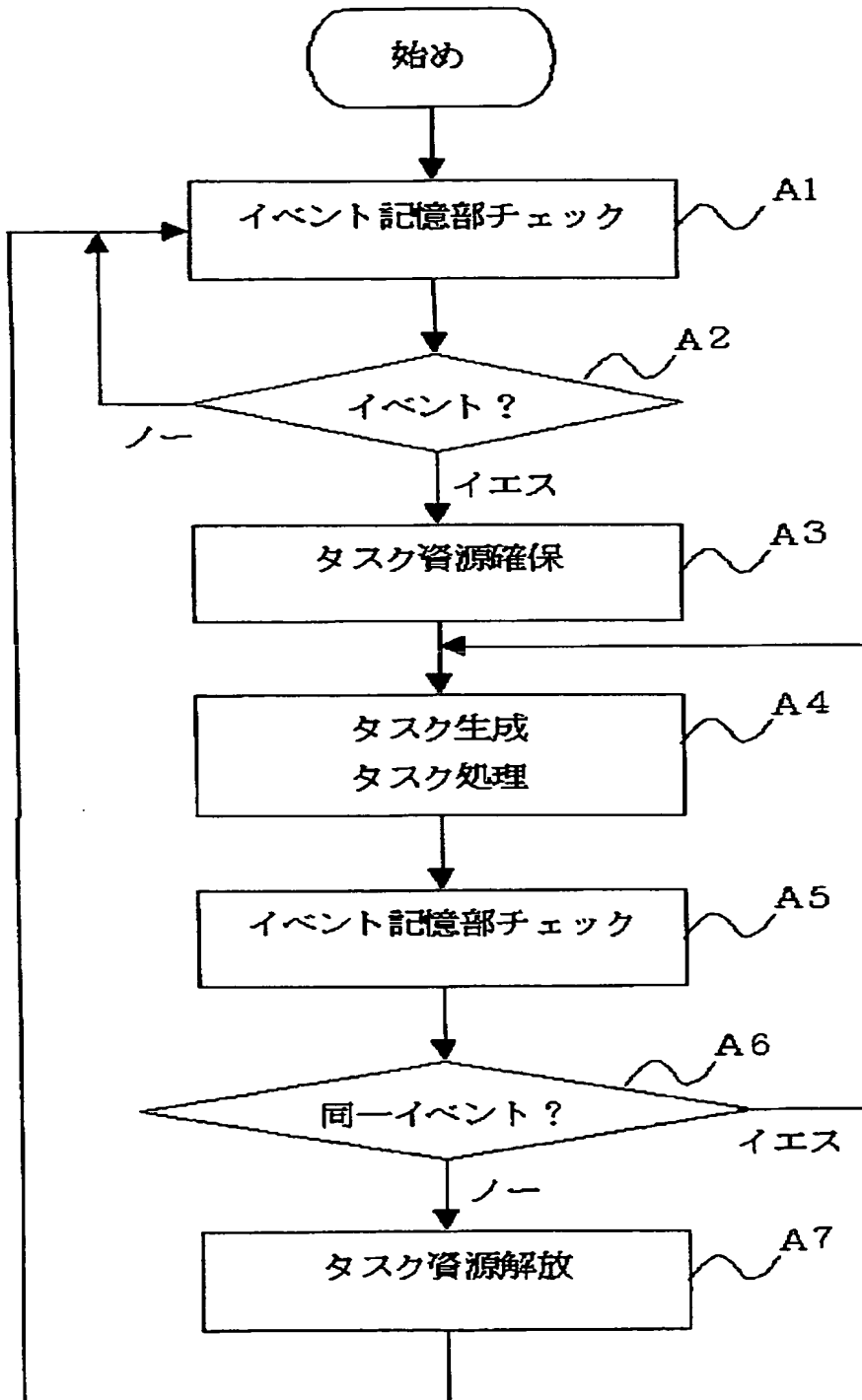
- 1 イベント発生装置
- 2 記憶装置
- 3 タスク制御装置
- 4、4 1、4 2、4 N タスク処理装置
- 2 1 イベント記憶部
- 2 2 タスク資源記憶部
- 2 3 実行中タスク記憶部
- 3 1 イベントチェック手段
- 3 2 タスク生成手段
- 3 3 タスク資源管理手段
- 2 1 1、2 3 1 イベント識別子

【書類名】 図面

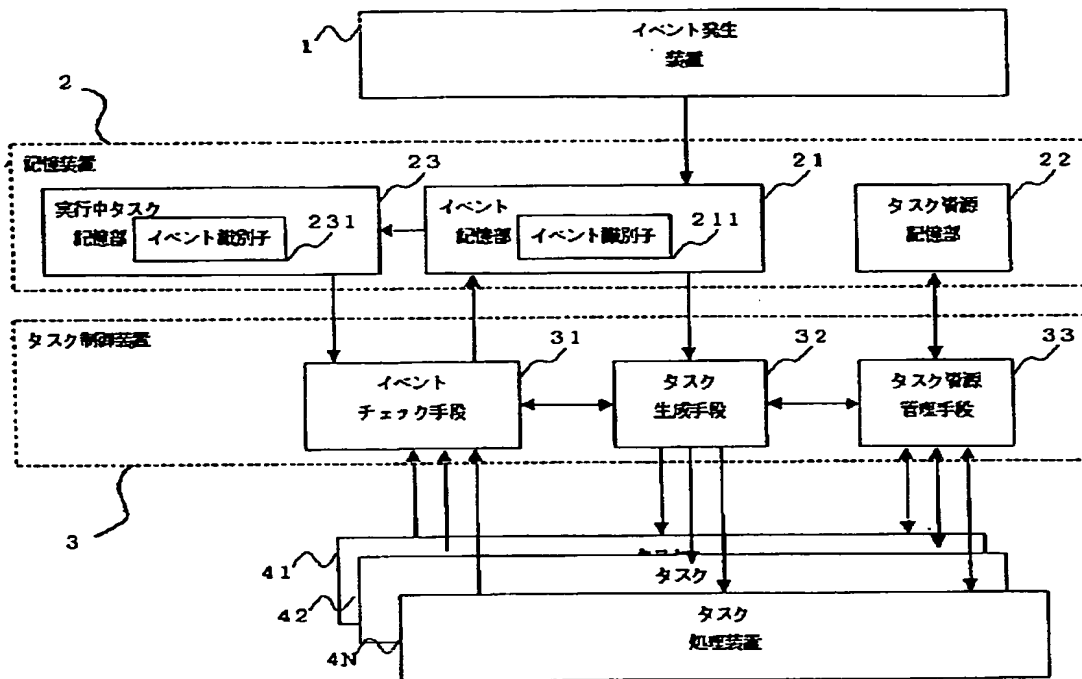
【図 1】



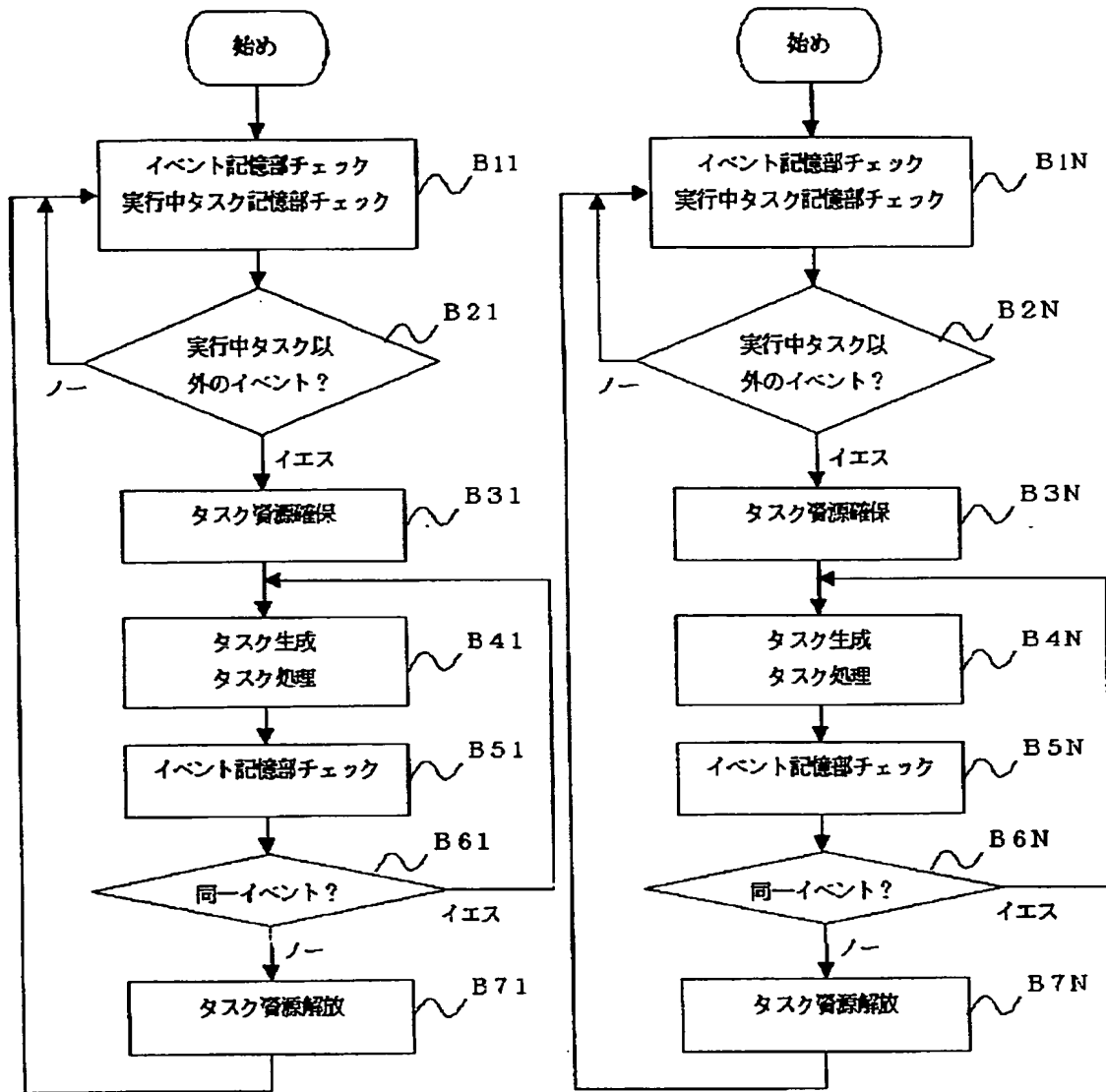
【図 2】



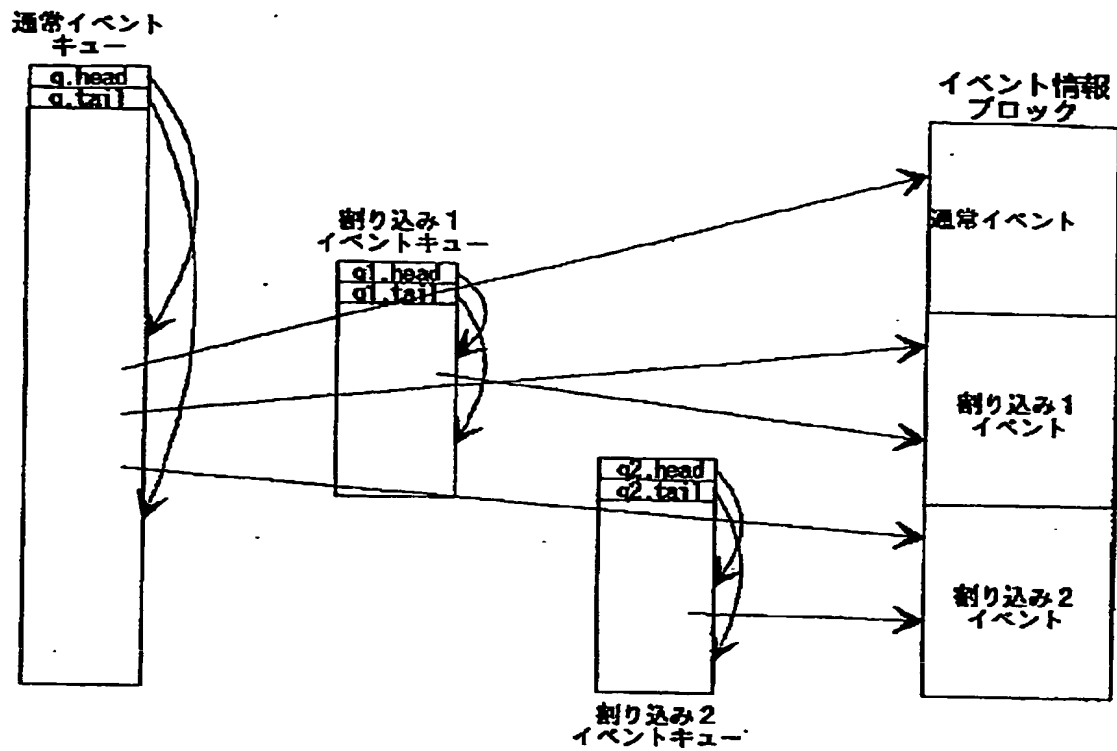
【図 3】



【図 4】



【図 5】



【書類名】 要約書

【要約】

【課題】 小規模なタスクを多量に処理する場合のタスク切り替えのためのオーバーヘッドの発生を抑制して、タスク処理速度を向上させる。

【解決手段】 タスク処理装置 4 でタスク処理が完了した時点で、イベントチェック手段 31 により、完了したタスクを生成したイベントと同一種類のイベントがイベント記憶部 21 に登録されているかが検査される。登録されていない場合、完了したタスクが必要とした資源のうち、タスク資源記憶部 22 への書き戻しの必要な資源がタスク資源管理手段 33 により戻される。他の種類のイベントがイベント記憶部 21 に登録されている場合、タスク生成および資源の確保が繰り返される。完了したタスクを生成したイベントと同一種類のイベントがイベント記憶部 21 に登録されている場合、資源の解放および再確保は行われず、引き続き、同一タスクがタスク処理装置 4 で処理される。

【選択図】 図 1



出 願 人 履 歴 情 報

識別番号 [000004237]

1. 変更年月日 1990年 8月29日  
[変更理由] 新規登録  
住 所 東京都港区芝五丁目7番1号  
氏 名 日本電気株式会社